Search-Based Software Testing (SBST) ICSE Workshop Proposal

Phil McMinn (primary contact)
Department of Computer Science
University of Sheffield
Regent Court, Portobello, S1 4DP, UK
p.mcminn@sheffield.ac.uk

Mark Harman
Department of Computer Science
University College London
Malet Place, London, WC1E 6BT, UK
mark.harman@ucl.ac.uk

PROPOSAL ABSTRACT

Search-Based Software Testing (SBST) is a form of Search-Based Software Engineering (SBSE) that optimizes testing through computational search. SBST has matured and now needs to draw in the expertise of the wider software engineering community: combining SBST with wider software engineering goals will improve both testing and the wider community. This is uniquely possible through SBST because SBST's fitness function (a metric) can act as a common "lingua franca". Such metrics and fitness functions exist across the software engineering spectrum, and are ripe for use in combination with SBST. SBST has been successfully run since 2008 as a co-located workshop of ICST, a testing conference. Relocating SBST as an ICSE workshop will maximise the potential for this interaction with the wider SE community. There is abundant evidence that it will be intellectually productive and well received: 30 papers submitted to the main ICSE 2014 conference were concerned with SBSE (6% of all submissions). The ICSE community is thereby indicating its interest and willingness to interact with computational search. We have already obtained $1,500 funding to support the workshop and have two outstanding keynotes (Lionel Briand and Cristian Cadar) already signed up. This provides further evidence that the workshop will be successful and productive for ICSE 2014.

1. DESIRED LENGTH AND DATES

We propose to hold the workshop over two days, our preference would be to have the workshop scheduled before the main conference.

2. THEMES AND GOALS

Search-Based Software Testing (SBST) is one example of the application of search-based optimization techniques to problems in software engineering. There is a growing realization that optimization can be applied to all aspects of the software development process, its products and processes—a research area known as Search-Based Software Engineering (SBSE) [17]. SBST [21] is the process of using search-based optimization algorithms to specifically address problems in software testing.

Work in SBST has developed to the point at which it is now ripe for combination with other areas of software engineering. One example is requirements optimization [34]. Possibilities exist to optimize the selection of requirements based not only on traditional aspects of SBSE for requirements (customer satisfaction, cost etc.), but also on the implications for regression testing (coverage achievable, test execution time). Another opportunity is the incorporation of SBSE approaches to project planning [3, 4, 9] into a combined approach to optimized requirements, project planning and testing. That is, instead of merely discussing requirements choices devoid of their technical and managerial consequences, an automated approach could be employed where the implications of requirement choices on the project plan—used to staff and implement the requirements—are checked in advance. This could further impact regression testing, where multi-objective solutions may be sought that balance the competing objectives of requirement choices with implementation plans and efficient and effective regression tests.

The common "lingua franca" that makes these combinations possible is the definition of a fitness function, and the ability to optimize many fitness functions at once using multi-objective search techniques. A fitness function is merely a form of metric [14], and metrics exist right across the software engineering spectrum.

Further specific topics to be explored will include:

Non-Functional Testing and Other Test Adequacy Criteria. SBST just needs a fitness function, which can be used to assess (for example) non-functional criteria as well as conventional white-box structural coverage. Possible applications include testing memory consumption, power usage, execution time etc.

Multi-Objective Solutions. SBST can use many fitness functions at once, presenting the end-user with a choice of non-dominating solutions at the so-called "Pareto Front". Multi-objective optimization may be used to produce test suites that aim to satisfy multiple goals, such as generating test suites that assess worst-case performance as well as covering all the branches of a program.

Co-evolving Tests and Software Systems. Future SBST techniques may utilize co-evolution in a virtuous cycle whereby tests are generated to find faults that are then automatically fixed—interfacing with recent work presented at ICSE and elsewhere on automatic patch generation for faults using Genetic Programming [30].

Human Aspects and Integration into Real Test Environments. Search-based techniques present the possibility to take into account domain knowledge [22] or optimize tests for human usage (for example, incorporating a language model to produce readable test strings, aiding human comprehension of the tests generated [1]). Other areas of SBSE have also involved a human-in-the-loop to steer the search process towards subjective goals [6].

Application of SBST in Industrial Practice. Lionel Briand (University of Luxembourg) has agreed to give a keynote talk that relates his experience of industrial projects and where SBST can and should be making an impact.

Combining SBST with Dynamic Symbolic Execution (DSE). Both SBST and DSE have been shown to be effective approaches for software testing. Each has complementary technical advantages that has motivated recent work on *combining* the two approaches [19, 5]. Cristian Cadar (Imperial College, UK) will give a keynote on this topic.

Abstract for ICSE Website:

Search-Based Software Testing (SBST) is the application of optimizing search techniques (for example, Genetic Algorithms) to solve problems in software testing. SBST is used to generate test data, prioritize test cases, minimize test suites, reduce human oracle cost, verify software models, test service-orientated architectures, construct test suites for interaction testing, and validate real-time properties.

The objectives of this workshop are to bring together researchers and industrial practitioners from both SBST and the wider software engineering community to share experience and provide directions for future research, and to encourage the use of search techniques to combine aspects of software testing with other aspects of the software engineering lifecycle.

3. RELEVANCE TO SOFTWARE ENGINEERING

SBST has been applied to a wide variety of testing goals including structural [18, 25], functional [28], non-functional [29] and state–based properties [12, 24]. Many approaches to testing and a wide diverse range of development domains have been addressed, including agents [23], aspects [15], exceptions [26], interactions [10], integration [11, 7], mutation [16, 20, 33], regression [27, 31, 32], stress [8, 13] and web applications [2].

In all approaches to SBST, the primary concern is to define a fitness function (or set of fitness functions) that capture the test objectives. The fitness function is used to guide a search-based optimization algorithm, which searches the space of test inputs to find those that meet the test objectives. Because any test objective can, in principle, be recast as a fitness function, the approach is highly generic and therefore widely applicable—as demonstrated by the prior list of testing applications. It is therefore highly relevant to Software Engineering. This relevance is further testified by the increasing cumulative number of publications in the area since 2007, as Figure 1 shows, providing evidence that SBST is a growing research area.

3.1 Relevance To ICSE

The SBST workshop compliments ICSE well. No fewer than 30 papers submitted to ICSE 2014 used some form of search technique coupled with a fitness function, accounting for 6% of submissions to the conference. This shows that there are already potential participants interested in SBSE who wish to attend ICSE. As stated previously, one of the reasons we wish to hold SBST with ICSE is our desire to reach out to other areas of software engineering expertise that may be able to produce mutual benefits with SBST.

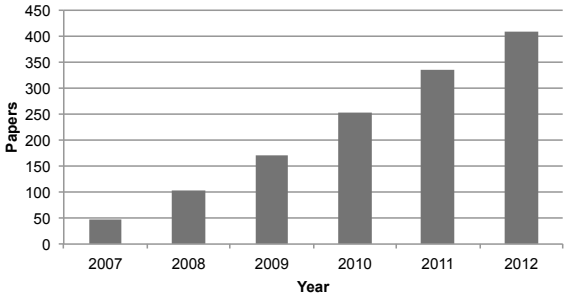

Figure 1: Publications in SBST since 2007

3.2 Balance and Synergy Across ICSE Events

Furthermore, SBST complements workshops regularly appearing at ICSE, including the most closely related workshop—the Automated Software Test (AST) workshop. We believe this is for two reasons:

1) SBST is More Than Just Automation. One of the benefits of the SBST approach is the ability to include a human-in-the-loop and to involve their domain expertise. This has already been exploited in search-based test data generation, where manually-produced test cases have been used to "seed" the search process. Other techniques have used resources produced from man-made artefacts, including models of natural language [1]. Furthermore, search-based approaches allow for humans to supply judgements and act as a fitness function in their own right [6].

2) Search-Based Optimization Techniques are Specialised. Optimization is a research field in itself, and as such the application of techniques from this domain require specific attention. The incorporation of an optimization technique involves thinking about the problem representation such that problem instances can be adapted by a search process, that search operators like crossover and mutation are properly tuned so that an optimization algorithm will not converge too early yet will not degenerate into a random search, etc. As such, the specifics of applying search-based optimization to testing are unlike the issues encountered in other areas of automated software engineering. Therefore, we as a community believe a dedicated workshop capable of discussing these matters is of importance and interest to ICSE participants, and as such, SBST a useful addition to the ICSE set of workshops.

4. PLANS FOR GENERATING DISCUSSION

Our plans to generate and stimulate discussion at the workshop aim to foster the generation of ideas for further interesting and stimulating future research in SBST, as well as encouraging potential future working relationships between participants. The starting point for discussions will be the following components:

Who's Who Session. Each participant will be asked to present a slide about their current research and what they

hope to get out of the workshop. This activity will serve as an "ice-breaker", while ensuring that participants get to know each other from the beginning of the event.

Keynotes. Lionel Briand (University of Luxembourg) will talk on applying SBST in industry, while **Cristian Cadar** (Imperial College, UK) will give a keynote on combining DSE with SBST. Following time for questions, we will ask the keynotes to raise three issues arising from their talk for participants to discuss in small groups. Each group will nominate somebody to feed back to the room. Discussion points will later be transcribed onto the workshop website.

Paper Presentation Sessions. Each paper will be given a 30 minute slot of which no more than 20 minutes will be devoted to the actual presentation. The final 10 minutes will be allotted for questions and discussion. We will nominate a "discussant" for each paper who will prepare questions in advance and be responsible for generating discussion points. To facilitate this, papers will be distributed in advance of the workshop and participants will be encouraged to read them in advance. We will also run a PhD paper session to enable the encouragement and coaching of doctoral students researching SBST. Each doctoral student will be assigned a "mentor" of high standing in the community to discuss the student's work and provide further feedback after their talk. The mentor will also be available for advice on further issues.

"Future of SBST" Discussion Panel Session. We will have a discussion panel of software engineering experts to discuss the future of SBST. Each panel member will open with their position statement, and questions will be invited from the audience. Gordon Fraser (University of Sheffield, UK) has agreed to chair the panel. The following have agreed in principle to be panelists, subject to travel constraints nearer the time:

- Darko Marinov (*University of Illinois at Urbana-Champaign, USA*)
- Alex Orso (*Georgia Institute of Technology, USA*)
- Paolo Tonella (*Fondazione Bruno Kessler–IRST, Italy*)
- Corina Pasareanu (*Carnegie Mellon University, USA*)
- Andreas Zeller (*Saarland University, Germany*)

We plan to hold the workshop over 2 days, since we expect to accept approximately 15 papers, including 3 PhD track papers. An indicative draft programme is as follows:

Day 1:	8:45 – 9:00am	Introduction
	9:00 – 10:00am	"Who's Who" session
	10:00 – 11:30am	Keynote: Lionel Briand
	11:30 – 11:45am	*Break*
	11:45 – 1:15pm	Paper session 1 (3 papers)
	1:15 – 2:00pm	*Lunch*
	2:00 – 3:30pm	Panel—"The Future of SBST"
	3:30 – 3:45pm	*Break*
	3:45 – 5:15pm	Paper session 2 (3 papers)
Day 2:	9:00 – 10:30am	Keynote: Cristian Cadar
	10:30 – 10:45am	*Break*
	10:45 – 12:15pm	PhD paper session (3 papers)
	12:15 – 1:00pm	*Lunch*
	1:00 – 2:30pm	Paper Session 3 (3 papers)
	2:30 – 2:45pm	*Break*
	2:45 – 4:15pm	Paper Session 4 (3 papers)
	4:15pm	Close

5. WEBSITE

URL: `http://www.searchbasedsoftwaretesting.org`
Our appointed web chair, Sina Shamshiri (PhD candidate, University of Sheffield), has designed our website for easy viewing on a mobile/tablet device as well as a desktop, so that it is usable by participants physically present at ICSE and the SBST workshop.

6. PROGRAM COMMITTEE MEMBERS

The following people will be on the programme committee (all confirmed). They have been chosen because they have presented high quality work in the past on SBST:

- Rob Alexander (*University of York, UK*)
- Giuliano Antoniol (*École Polytechnique de Montréal*)
- Andrea Arcuri (*Simula Research Labs, Norway*)
- Francisco Chicano (*University of Málaga*)
- Myra Cohen (*University of Nebraska at Lincoln, USA*)
- Massimiliano Di Penta (*University of Sannio, Italy*)
- Gordon Fraser (*University of Sheffield, UK*)
- Yvan Labiche (*Carleton University, Canada*)
- Gregory Kapfhammer (*Allegheny College, USA*)
- Simon Poulding (*University of York, UK*)
- Paolo Tonella (*Fondazione Bruno Kessler–IRST, Italy*)
- Tanja Vos (*Universidad Politecnica de Valencia*)
- Joachim Wegener (*Berner & Mattner, Germany*)
- Westley Weimer (*University of Virginia, USA*)
- David White (*University of Glasgow, UK*)
- Shin Yoo (*University College London, UK*)
- Andreas Zeller (*Saarland University, Germany*)

7. PARTICIPANT SOLICITATION

We intend to run an open workshop, in which papers will be solicited for presentation. We plan to make a call for full papers, short papers and PhD papers. Each paper will undergo a review process, receiving three reviews from members of the programme committee—who will declare conflicts in advance and bid for the papers they are interested in. Following reviews, there will be an online discussion process, and finally the organisers will make final decisions on paper acceptance, using referee reviews and conclusions of the discussions. Participation is not limited to presenters—workshop attendance will be open to all.

8. PAPERS AND PROCEEDINGS PLANS

We would expect approximately 15 papers compromising of full papers (10 pages in length), short papers (4 pages in length) and PhD papers (also 4 pages in length). Accepted papers will appear in a pre-proceedings and will be proposed for publication in the ACM Digital Library.

9. EXPECTED NUMBER OF PARTICIPANTS

SBST has been successfully held at ICST, a testing conference, since 2008. The event could easily continue at ICST, however we wish to grow the workshop and the SBST field by reaching out to the wider software engineering community. We believe ICSE is the ideal conference for this. As stated in Section 2, there is a unique opportunity for SBST to touch other areas of software engineering.

We expect approximately 35 participants to attend SBST if held at ICSE due to location at a large venue with a wide appeal, and because of our plans for the event including our prestigious keynotes and panel members. This judgement is also based on the fact that 6% (30) of ICSE papers featured a search-based optimization algorithm, and as such there is already a significant foundation of interest in the search-based field at ICSE.

10. LOGISTICS

We envisage that we will only require a projector and projector screen for the presenters. Presenters will be able to use their own laptops, but will be able to use an organiser's, if necessary. A flipboard and/or whiteboard would be very useful for recording salient discussion points.

11. ORGANIZER BACKGROUND

Phil McMinn is a Senior Lecturer (Associate Professor) at the University of Sheffield, UK and is the Director for Software Testing at the University's Advanced Computing Research Centre. His main research interest lies in SBST, and his survey of the field in 2004 [21] has amassed over 700 citations according to Google Scholar. He founded the SBST workshop in 2008, which was established at ICST. He was general chair of the Symposium of Search-Based Software Engineering (SSBSE) in 2011. Prior to this, he was program chair of TAIC PART in 2006 and 2007, and the UK testing workshop in 2005. He was track chair of the SBSE track at GECCO 2006. He was workshops chair for the ICST conference in 2011, which attracted a record 12 successful workshops. In total he has served on over 30 conference and workshop programme committees including ICSE, ISSTA and ICST. He is one of the founding steering committee members for SSBSE. He has co-edited three journal special issues on Software Testing for Software Testing, Verification and Reliability and the Journal of Systems and Software.

Mark Harman is Professor of Software Engineering at University College London (UCL) and director of the Centre for Research in Evolution Search and Testing (CREST) within the Department of Computer Science at UCL. He is a recognised world leader in the area of Search Based Software Engineering (SBSE), having given 24 keynotes on SBSE-related topics at international academic conferences and workshops (including ETAPS, ASE and ESEM). He also gave a widely-cited ICSE Future of Software Engineering presentation on SBSE in 2007. His SBSE work is used by ABB, Daimler, Ericsson, Google, IBM, Microsoft and Motorola. A recent [6] ten year retrospective on SBSE work since the term was coined in 2001 by Harman and Jones, ranked Harman first place among the (800 plus) authors in the field of SBSE, and reported that 5 of his papers occupied positions in the top 10 papers by citation count (according to multiple citation measures).

12. FUNDING AND SPONSORSHIP

We have already secured funding of $1,500 and will seek further sponsorship if accepted. Funding will be used to support keynotes and panelist attendance, with an extra night at the ICSE hotel. This partial funding for attendees will encourage registration for ICSE itself.

13. REFERENCES

[1] S. Afshan, P. McMinn, and M. Stevenson. Evolving readable string test inputs using a natural language model to reduce human oracle cost. In ICST, 2013.

[2] N. Alshahwan and M. Harman. Automated web application testing using search based software engineering. In ASE, 2011.

[3] G. Antoniol, M. Di Penta, and M. Harman. The use of search-based optimization techniques to schedule and staff software projects: An approach and an empirical study. *Software — Practice and Experience*, 2011.

[4] G. Antoniol, S. Gueorguiev, and M. Harman. Software project planning for robustness and completion time in the presence of uncertainty using multi objective search based software engineering. In GECCO, 2009.

[5] A. Baars, M. Harman, Y. Hassoun, K. Lakhotia, P. McMinn, P. Tonella, and T. Vos. Symbolic search-based testing. In ASE, 2011.

[6] G. Bavota, F. Carnevale, A. De Lucia, M. Di Penta, and R. Oliveto. Putting the developer in-the-loop: an interactive ga for software re-modularization. In SSBSE, 2012.

[7] L. C. Briand, J. Feng, and Y. Labiche. Using genetic algorithms and coupling measures to devise optimal integration test orders. In SEKE, 2002.

[8] L. C. Briand, Y. Labiche, and M. Shousha. Stress testing real-time systems with genetic algorithms. In GECCO, 2005.

[9] F. Chicano and E. Alba. Management of software projects with gas. In MIC, 2005.

[10] M. B. Cohen, P. B. Gibbons, W. B. Mugridge, and C. J. Colbourn. Constructing test suites for interaction testing. In ICSE, 2003.

[11] T. Colanzi, W. Assuncao, S. Vergilio, and A. Pozo. Integration test of classes and aspects with a multi-evolutionary and coupling-based approach. In SSBSE, 2011.

[12] K. Derderian, R. Hierons, M. Harman, and Q. Guo. Automated Unique Input Output sequence generation for conformance testing of FSMs. *The Computer Journal*, 2006.

[13] C. D. Grosso, G. Antoniol, M. D. Penta, P. Galinier, and E. Merlo. Improving network applications security: a new heuristic to generate stress testing data. In GECCO, 2005.

[14] M. Harman and J. Clark. Metrics are fitness functions too. In METRICS, 2004.

[15] M. Harman, F. Islam, T. Xie, and S. Wappler. Automated test data generation for aspect-oriented programs. In AOSD, 2009.

[16] M. Harman, Y. Jia, and B. Langdon. Strong higher order mutation-based test data generation. In ESEC/FSE, 2011.

[17] M. Harman and B. F. Jones. Search based software engineering. *Information and Software Technology*, 2001.

[18] M. Harman and P. McMinn. A theoretical and empirical study of search based testing: Local, global and hybrid search. *IEEE TSE*, 2010.

[19] K. Inkumsah and T. Xie. Improving structural testing of object-oriented programs via integrating evolutionary testing and symbolic execution. In ASE, 2008.

[20] Y. Jia and M. Harman. Higher order mutation testing. *Journal of Information and Software Technology*, 2009.

[21] P. McMinn. Search-based software test data generation: A survey. *Software Testing, Verification and Reliability*, 2004.

[22] P. McMinn, M. Stevenson, and M. Harman. Reducing qualitative human oracle costs associated with automatically generated test data. In STOV, 2010.

[23] C. Nguyen, A. Perini, P. Tonella, S. Miles, M. Harman, and M. Luck. Evolutionary testing of autonomous software agents. In AAMAS, 2009.

[24] J. Oh, M. Harman, and S. Yoo. Transition coverage testing for Simulink/Stateflow models using messy genetic algorithms. In GECCO, 2011.

[25] P. Tonella. Evolutionary testing of classes. In ISSTA, 2004.

[26] N. Tracey, J. Clark, K. Mander, and J. McDermid. Automated test-data generation for exception conditions. *Software Practice and Experience*, 2000.

[27] K. R. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. S. Roos. Time aware test suite prioritization. In ISSTA, 2006.

[28] J. Wegener and O. Bühler. Evaluation of different fitness functions for the evolutionary testing of an autonomous parking system. In GECCO, 2004.

[29] J. Wegener and M. Grochtmann. Verifying timing constraints of real-time systems by means of evolutionary testing. *Real-Time Systems*, 1998.

[30] W. Weimer, T. Nguyen, C. L. Goues, and S. Forrest. Automatically Finding Patches Using Genetic Programming. In ICSE, 2009.

[31] S. Yoo and M. Harman. Pareto efficient multi-objective test case selection. In ISSTA, 2007.

[32] S. Yoo, M. HarmInan, P. Tonella, and A. Susi. Clustering test cases to achieve effective and scalable prioritisation incorporating expert knowledge. In ISSTA, 2009.

[33] Y. Zhan and J. A. Clark. Search-based mutation testing for simulink models. In GECCO, 2005.

[34] Y. Zhang, A. Finkelstein, and M. Harman. Search based requirements optimisation: Existing work and challenges. In REFSQ, 2008.

call for papers

(subject to workshop being accepted)

search based software testing 2014

Held in conjunction with ICSE 2014 - IEEE International Conference on Software Engineering

May 31—June 7, 2014
Hyderabad, India

Keynote Speakers

Lionel Briand
Professor and FNR PEARL Chair, Interdisciplinary Centre for ICT Security, Reliability and Trust (SnT) University of Luxembourg

Cristian Cadar
Senior Lecturer in the Department of Computing at Imperial College London, where he leads the Software Reliability Group

About the workshop:

Search-Based Software Testing (SBST) is the application of optimizing search techniques (for example, Genetic Algorithms) to solve problems in software testing. SBST is used to generate test data, prioritize test cases, minimize test suites, reduce human oracle cost, verify software models, test service-orientated architectures, construct test suites for interaction testing, and validate real-time properties.

The objectives of this workshop are to bring together researchers and industrial practitioners both from SBST and the wider software engineering community to share experience and provide directions for future research, and to encourage the use of search techniques to combine aspects of software testing with other aspects of the software engineering lifecycle.

Program Committee

Rob Alexander (University of York, UK) **Giuliano Antoniol** (École Polytechnique de Montréal)
Andrea Arcuri (Simula Research Labs, Norway) **Francisco Chicano** (University of Málaga)
Myra Cohen (University of Nebraska at Lincoln, USA) **Massimiliano Di Penta** (University of Sannio, Italy)
Gordon Fraser (University of Sheffield, UK) **Gregory Kapfhammer** (Allegheny College, USA)
Yvan Labiche (Carleton University, Canada) **Simon Poulding** (University of York, UK)
Paolo Tonella (Fondazione Bruno Kessler–IRST, Italy) **Tanja Vos** (Universidad Politecnica de Valencia)
Joachim Wegener (Berner & Mattner, Germany) **Westley Weimer** (University of Virginia, USA)
Shin Yoo (University College London, UK) **Andreas Zeller** (Saarland University, Germany)

Researchers and practitioners are invited to submit:

* **Full papers** (maximum of 10 pages) to the workshop on original research--either empirical or theoretical--in SBST, practical experience of using SBST, or SBST tools.

* **Short papers** (maximum of 4 pages) that describe novel techniques, ideas and positions that have yet to be fully developed; or are a discussion of the importance of a recently published SBST result by another author in setting a direction for the SBST community, and/or the potential applicability (or not) of the result in an industrial context

* **PhD papers** (maximum of 4 pages). PhD papers are invited for students to showcase their research and to receive feedback from senior members of the SBST community. See the website for eligibility constraints and requirements.

In all cases, papers should address a problem in the software testing/verification/validation domain or combine elements of those domains with other concerns in the software engineering lifecycle. Examples of problems in the software testing/verification/validation domain include (but are not limited to) generating testing data, prioritizing test cases, minimizing test suites, verifying software models, testing service-orientated architectures, constructing test suits for interaction testing, and validating real-time properties.

The solution should apply a metaheuristic search strategy such as (but not limited to) random search, local search (e.g. hill climbing, simulated annealing, and tabu search), evolutionary algorithms (e.g. genetic algorithms, evolution strategies, and genetic programming), ant colony optimization, and particle swarm optimization.

Format and Submission:

All papers must conform, at time of submission, to the ACM Formatting Guidelines (LaTeX users, please use the "Option 2" style). All submissions must be in PDF format, and submitted to [details to appear nearer the time]

Important Dates:

[standard dates stated on ICSE website http://2014.icse-conferences.org/workshops]

Workshop Chairs:

Phil McMinn
University of Sheffield, UK
(p.mcminn@sheffield.ac.uk)

Mark Harman
University College London, UK
(mark.harman@ucl.ac.uk)

WWW.SEARCHBASEDSOFTWARETESTING.ORG